

**mục lục**

[Câu 1: các định lý cơ bản của đại số Boolean.....1](#)

[Câu 2 : các công thức logic cơ bản.....2](#)

[Câu 3: đơn giản biểu thức.....4](#)

[Câu 4: thiết kế mạch logic .....5](#)

[Câu 5 các mạch mã hóa.....9](#)

[Câu 6: các mạch giải mã.....12](#)

[Câu 7: mạch hợp kênh và phân kênh.....14](#)

[Câu 8: Các mạch số học.....17](#)

[Câu 9: Các mạch Flip - Flop cơ bản .....23](#)

[Câu 10: các loại mạch đếm.....31](#)

[Câu 11: mạch chốt và ghi dịch.....38](#)

[Câu 12: bộ biến đổi DAC .....42](#)

[Câu 13: bộ biến đổi ADC.....45](#)

[Câu 14: mạch tạo xung dùng 2 cổng NAND.....56](#)

[Câu 15: Mạch tạo xung dùng IC 555.....57](#)

**Câu 1: các định lý cơ bản của đại số Boolean**  
**+ các mệnh đề cơ sở**

$$X + \quad = 1$$

$$X \cdot \quad = 0$$

$$X + 1 = 1$$

$$X \cdot 1 = X$$

+ **Định luật hấp thụ** được một số trang đầu. Vui lòng download file gốc để xem toàn bộ các trang

$$X + X = X$$

$$X \cdot X = X$$

+ **Định luật phủ định của phủ định.**

$$\overline{\overline{X}} = X$$

+ **Định luật kết hợp**

$$X_1 + (X_2 + X_3) = (X_1 + X_2) + X_3$$

$$(X_1 \cdot X_2) \cdot X_3 = X_1 \cdot (X_2 \cdot X_3)$$

+ **định luật giao hoán.**

$$X_1 + X_2 = X_2 + X_1$$

$$X_1 \cdot X_2 = X_2 \cdot X_1$$

+ **Định luật phân phối**

$$X_1 \cdot (X_2 + X_3) = X_1 X_2 + X_1 X_3$$

$$(X_1 + X_2)(X_1 + X_3) = X_1 + X_2 \cdot X_3$$

+ **Định luật DEMORGAN**

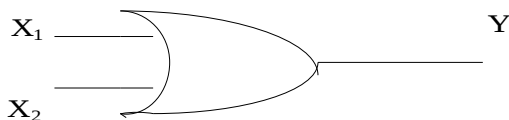
$$\overline{X_1 X_2} = \overline{X_1} + \overline{X_2}$$

$$\overline{X_1 + X_2} = \overline{X_1} \cdot \overline{X_2}$$

**Câu 2 : các công thức logic cơ bản**

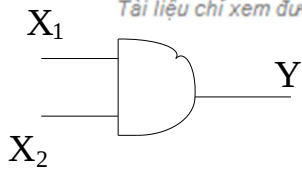
+ **phép cộng logic - hàm hoặc (OR)**

$$Y = X_1 + X_2$$



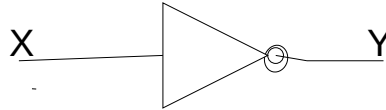
+ **Phép nhân logic - hàm và (AND)**

$$Y = X_1 \cdot X_2$$



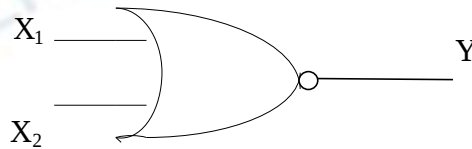
### + Phép phủ định - hàm đảo ( NOT)

$Y =$



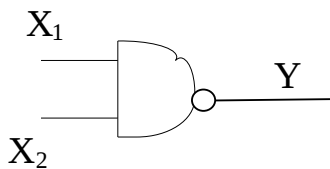
### + Hàm không hoặc ( NOR)

$$Y = \overline{X_1 + X_2}$$



### + Hàm không và ( NAND)

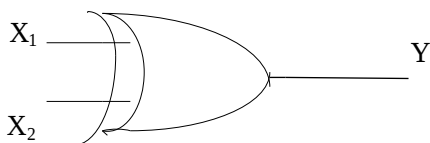
$$Y = \overline{X_1 \cdot X_2}$$



### + Hàm hoặc tuyệt đối 2 đầu vào

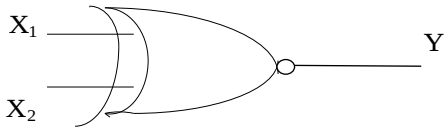
$$Y = X_1 \cdot 2 + 1 \cdot X_2$$

Được viết lại là:  $y = X_1 \oplus X_2$



### + Hàm không hoặc tuyệt đối (xnor)

$$Y = \overline{X_1 \oplus X_2}$$



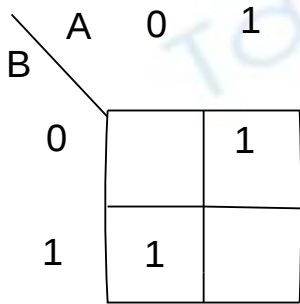
**Câu 3: đơn giản biểu thức**

**\* trường hợp xây dựng hàm logic theo phương pháp giải tích:**

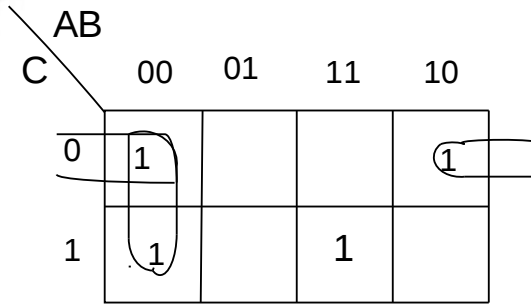
- ta áp dụng các định luật của đại số logic để đơn giản hàm logic sao cho hàm cuối cùng là tối giản, thực hiện hàm cần ít phần tử logic cơ bản nhất

**\* trường hợp xây dựng hàm logic từ bảng karnaugh**

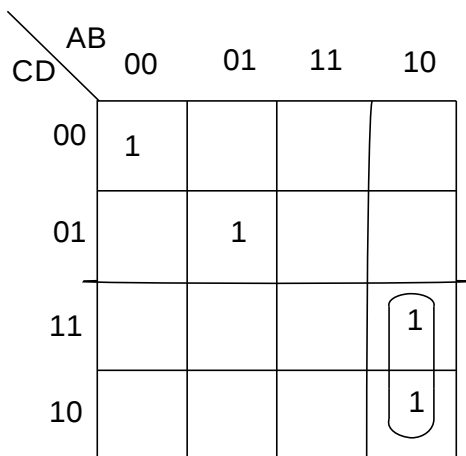
- ta hãy ghép các minterm ứng với  $f_1 = 1$  (các ô có số 1) ở các ô kề nhau theo hàng ngang hoặc hàng dọc và ghép các ô bằng 1 nằm đối diện nhau trong bảng như các đường khoanh vòng tròn trong hình 1. các minterm được ghép như vậy nhất định sẽ có thừa số chung và sẽ đơn giản được 1 biến bù nhau.



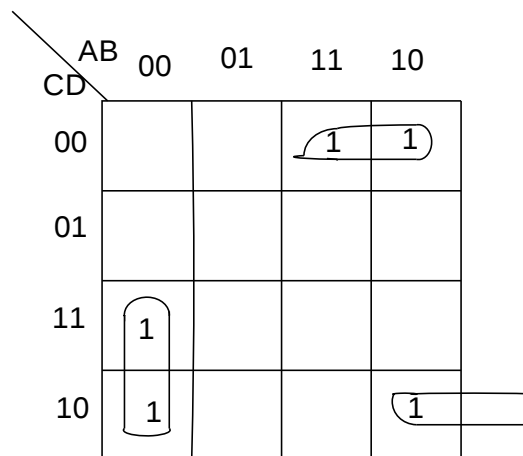
$F = B + A$   
hình a,



$F = + C + A + A B C$   
hình b,



$F = +BD+ACD+AC$        $F = AB+A+CD+C+AC$   
hình C,



hình d,

- hình b ta có:

$F = + C + A + A B C$       ghép 2 minterm kề nhau

$F = (+C) + C + A + ABC$       vì  $+C = 1$

$F = + A + ABC$

Để đơn giản 2 ô đối diện, ở đây ta viết thêm số hạng và hàm F vẫn không thay đổi ( vì + )= ta có:

$F = + A B C + A +$  ghép 2 số hạng cuối:

$F = + A B C + ( A + )$  cuối cùng:

$F = + A B C +$

- Hình d ta có:

Hàm F có số hạng C cộng thêm C vẫn không đổi:

$F = AB + A + CD + C + C$

$F = A (B + ) + C (D + ) + C (A + )$

$F = A + C + C$

#### Câu 4: thiết kế mạch logic

- để thiết kế các logic tổ hợp ta thực hiện các bước sau:

+, B<sub>1</sub> : từ yêu cầu chức năng ta lập bảng chân lý của hàm logic.

+, b<sub>2</sub> : từ bảng chân lý ta suy ra các phương trình logic bằng phương pháp minton hoặc matrix

+, b<sub>3</sub> tối giảm hàm logic có thể.

+, b<sub>4</sub> từ hàm logic ta có thể thiết kế được mạch logic bằng phần tử logic cơ bản

**1 ví dụ 1** : xây dựng mạch so sánh 2 số A B : A = B thì C = 1, A ≠ B thì C = 0

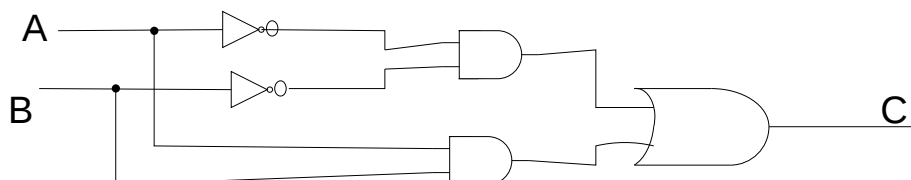
ta có bảng chân lý như hình dưới :

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

- từ bảng chân lý ta tìm được phương trình logic như sau:

$C = + AB$

Ở đây ta dùng phương pháp giải tích: lấy tổng các minterm ứng với C = 1 sơ đồ logic như sau:



- Thông thường ngày nay người ta hay dùng các bộ tổng hoặc bộ so sánh để thiết kế các cổng logic

+ Bộ tổng:

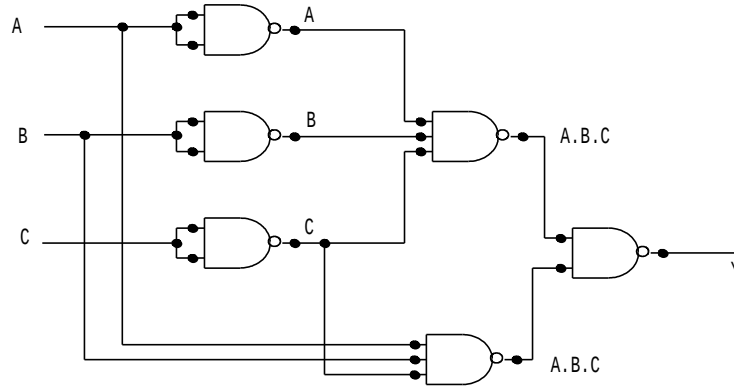
- bộ tổng bán phần ( half adder)

- bộ tổng toàn phần ( full adder : FA)



$$= \bar{A} \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C$$

- Dựa vào biểu thức boole ta vẽ mạch sau:



### 3. Ví dụ 3:

Cho hệ thức boole sau:

$$Y = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$$

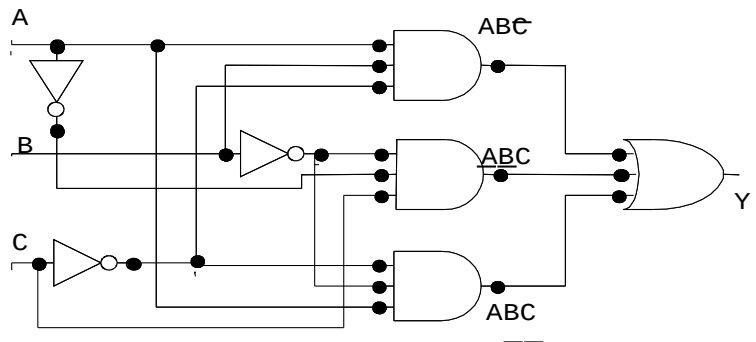
Hãy thiết lập mạch điện và bảng chọn lý thực hiện hàm sau:

Giải: Ta vẽ bảng chọn lý như sau:

$$T\bar{O} \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$$

ta vẽ mạch điện như sau:

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

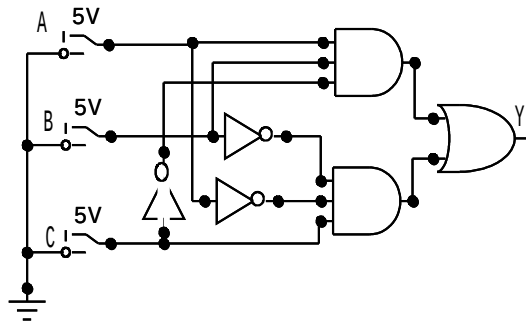


### 4. Ví dụ 4:

Cho hệ thức boole, hãy vẽ mạch điện và bảng chọn lý.

$$Y = A \cdot B \cdot C + \bar{A} \cdot B \cdot C$$

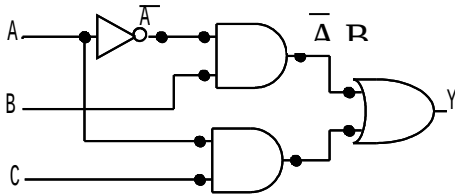
| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |



### 5 Ví dụ 5:

Cho mạch điện sau, hãy tìm lập bảng chọn lý và hệ thức boole?

ta cũ b<sup>1</sup>ng ch©n lý sau:



| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

### 6. VÝ d© 6:

- Mét tæ hìp sè nh<sup>1</sup>p ph©n gãm 4 bÝt sã cũ 16 tr<sup>1</sup>ng th<sup>1</sup>i cũa tæ hìp biÖn, h-y x©y dùng mét m<sup>1</sup>ch @iÖn @Ó x,c @Pnh c,c con sè tã 0000 @Ön 1001 lụ m-sè BCD - cũn c,c con sè kh<sup>1</sup>c kh«ng ph<sup>1</sup>i lụ m- BCD.

Ta qui íc nõu tæ hìp lụ m- BCD th<sup>1</sup>x hụm nhËn gi, tr<sup>1</sup>p 0 - @Çu ra cũ m¸c thÊp, cũn tæ hìp kh«ng ph<sup>1</sup>i lụ m- BCD hụm nhËn gi, tr<sup>1</sup>p 1 - @Çu ra cũ m¸c cao, ta cũ b<sup>1</sup>ng ch©n lý nh sau:

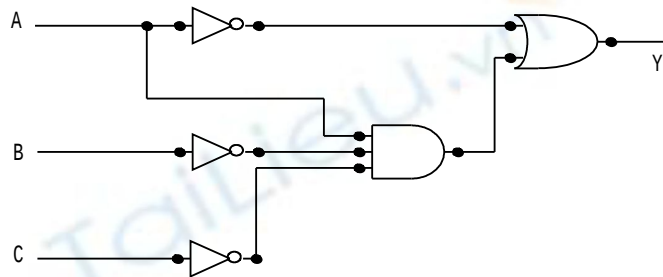
| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



Tổ hợp ch@n lý ta cã hÖ thøc boole nh sau:

$$\begin{aligned}
 Y &= \overline{A}.\overline{B}.\overline{C}.D + \overline{A}.\overline{B}.C.\overline{D} + \overline{A}.B.\overline{C}.\overline{D} + \overline{A}.B.C.D + \overline{A}.B.\overline{C}.D + \overline{A}.B.C.D + \\
 &\overline{A}.B.C.D + \overline{A}.B.C.D + \overline{A}.B.C.D + \overline{A}.B.C.D \\
 &= \overline{A}.\overline{B}.\overline{C} (D + \overline{D}) + \overline{A}.\overline{B}.C (\overline{D} + D) + \overline{A}.B.\overline{C} (D + \overline{D}) + \overline{A}.B.C (D + \overline{D}) + \\
 &\overline{A}.B.C (D + \overline{D}) \\
 &= \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + \overline{A}.B.\overline{C} + \overline{A}.B.C + \overline{A}.B.C \\
 &= \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + \overline{A}.B.\overline{C} + \overline{A}.B.C \\
 &= \overline{A}.B.(C + \overline{C}) + \overline{A}.\overline{B} (C + \overline{C}) + \overline{A}.B.\overline{C} = \overline{A}.B + \overline{A}.\overline{B} + \overline{A}.B.C \\
 &= \overline{A}(\overline{B} + B) + \overline{A}.\overline{B}.C = \overline{A} + \overline{A}.\overline{B}.C
 \end{aligned}$$

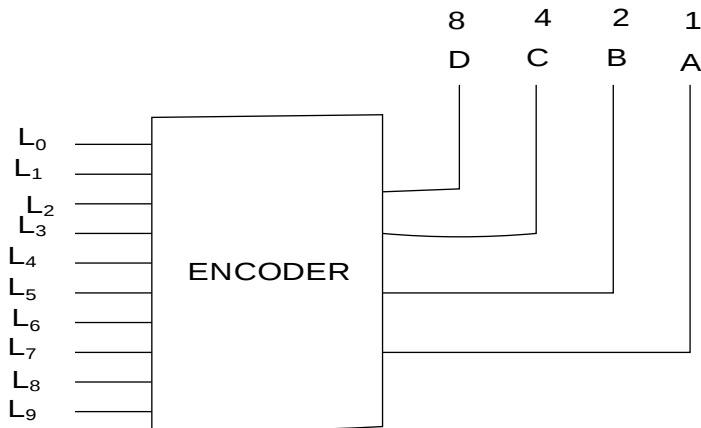
Tổ hÖ thøc boole ta cã m¹ch ®iÖn nh sau:



### Câu 5 các m¹nh mã hóa

#### \* chuyÖn từ mã thập phân sang mã BCD (Encoder)

- thiết bị có 10 lối vào  $L_0 \div L_9$  ứng với các số từ 0 ÷ 9 của hệ 10 và lối ra A,,B,C,D ứng với 1 từ nhị phân 4 bit. Sơ đồ khối như hình sau:



Hình 1: Sơ đồ khối mạch chuyÖn mã

Ta có bảng chuyÖn đổi mã như bảng 2. khi ấn phím thập phân  $L_x$  tức là đưa nó lên mức 1 thì lối ra BCD hiện lên từ nhị phân biểu diễn số  $L_x$  đó.

| Số Thập phân | Lối vào thập phân |                |                |                |                |                |                |                |                |                | BCD |   |   |   |
|--------------|-------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----|---|---|---|
|              | L <sub>0</sub>    | L <sub>1</sub> | L <sub>2</sub> | L <sub>3</sub> | L <sub>4</sub> | L <sub>5</sub> | L <sub>6</sub> | L <sub>7</sub> | L <sub>8</sub> | L <sub>9</sub> | D   | C | B | A |
| 0            | 1                 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0   | 0 | 0 | 0 |
| 1            | 0                 | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0   | 0 | 0 | 1 |
| 2            | 0                 | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0   | 0 | 1 | 0 |
| 3            | 0                 | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0   | 0 | 1 | 1 |
| 4            | 0                 | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0   | 1 | 0 | 0 |
| 5            | 0                 | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0   | 1 | 0 | 1 |
| 6            | 0                 | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0   | 1 | 1 | 0 |
| 7            | 0                 | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0   | 1 | 1 | 1 |
| 8            | 0                 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 1   | 0 | 0 | 0 |
| 9            | 0                 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 1   | 0 | 0 | 1 |

Bảng 2 bảng chuyển mã từ mã 10 sang mã BCD

- từ bảng chân lý ta tìm hàm lối ra bằng cách lấy tổng các minterm:

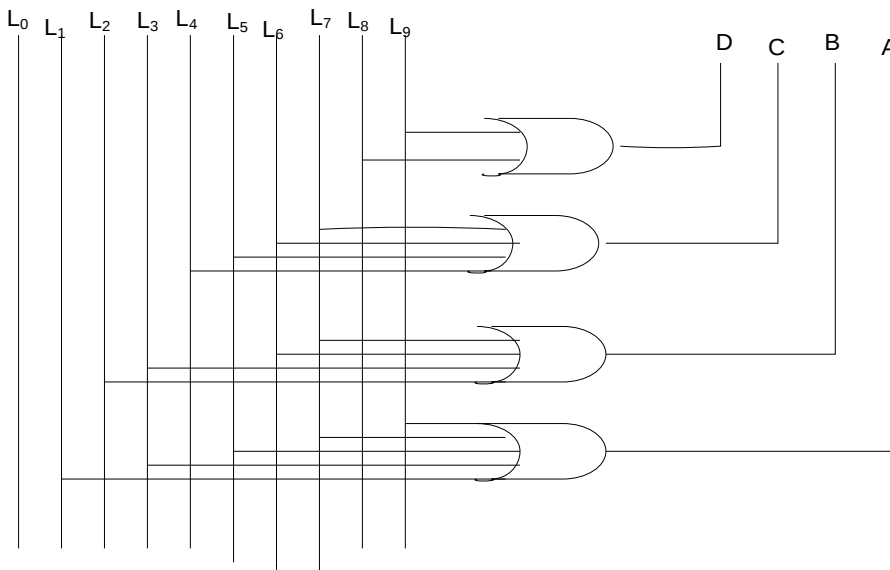
$$A = L_1 + L_3 + L_5 + L_7 + L_9$$

$$B = L_2 + L_3 + L_6 + L_7$$

$$C = L_4 + L_5 + L_6 + L_7$$

$$D = L_8 + L_9$$

- Từ đây có thể dùng mạch HOẶC (OR) nhiều lối vào để xây dựng mạch. Sơ đồ mạch mã hóa xây dựng được như hình sau:



\* chuyển từ mã nhị phân sang mã Gray.

- mã nhị phân đưa vào lối vào  $A_0, A_1, A_2, A_3$  lối ra nhận được mã Gray  $G_0, G_1, G_2, G_3$ .

sơ đồ khối như hình sau



Bảng chân lý như hình dưới đây:

| Số thập phân | Mã nhị phân |       |       |       | Mã Gray |       |       |       |
|--------------|-------------|-------|-------|-------|---------|-------|-------|-------|
|              | $A_3$       | $A_2$ | $A_1$ | $A_0$ | $G_0$   | $G_1$ | $G_2$ | $G_3$ |
| 0            | 0           | 0     | 0     | 0     | 0       | 0     | 0     | 0     |
| 1            | 0           | 0     | 0     | 1     | 0       | 0     | 0     | 1     |
| 2            | 0           | 0     | 1     | 0     | 0       | 0     | 1     | 1     |
| 3            | 0           | 0     | 1     | 1     | 0       | 0     | 1     | 0     |
| 4            | 0           | 1     | 0     | 0     | 0       | 1     | 1     | 0     |
| 5            | 0           | 1     | 0     | 1     | 0       | 1     | 1     | 1     |
| 6            | 0           | 1     | 1     | 0     | 0       | 1     | 0     | 1     |
| 7            | 0           | 1     | 1     | 1     | 0       | 1     | 0     | 0     |
| 8            | 1           | 0     | 0     | 0     | 1       | 1     | 0     | 0     |
| 9            | 1           | 0     | 0     | 1     | 1       | 1     | 0     | 1     |
| 10           | 1           | 0     | 1     | 0     | 1       | 1     | 1     | 1     |
| 11           | 1           | 0     | 1     | 1     | 1       | 1     | 1     | 0     |
| 12           | 1           | 1     | 0     | 0     | 1       | 0     | 1     | 0     |
| 13           | 1           | 1     | 0     | 1     | 1       | 0     | 1     | 1     |
| 14           | 1           | 1     | 1     | 0     | 1       | 0     | 0     | 1     |
| 15           | 1           | 1     | 1     | 1     | 1       | 0     | 0     | 0     |

\* **Chuyển từ mã nhị phân sang mã bù 2 nhị phân**

- ta có bảng chân lý như sau:

| Số thập phân | Mã nhị phân |       |       |       | Mã bù nhị phân |       |       |       |
|--------------|-------------|-------|-------|-------|----------------|-------|-------|-------|
|              | $A_3$       | $A_2$ | $A_1$ | $A_0$ | $B_3$          | $B_2$ | $B_1$ | $B_0$ |
| 0            | 0           | 0     | 0     | 0     | 0              | 0     | 0     | 0     |
| 1            | 0           | 0     | 0     | 1     | 1              | 1     | 1     | 1     |
| 2            | 0           | 0     | 1     | 0     | 1              | 1     | 1     | 0     |
| 3            | 0           | 0     | 1     | 1     | 1              | 1     | 0     | 1     |
| 4            | 0           | 1     | 0     | 0     | 1              | 1     | 0     | 0     |
| 5            | 0           | 1     | 0     | 1     | 1              | 0     | 1     | 1     |
| 6            | 0           | 1     | 1     | 0     | 1              | 0     | 1     | 0     |
| 7            | 0           | 1     | 1     | 1     | 1              | 0     | 0     | 1     |

|    |   |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|---|
| 8  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9  | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

**Câu 6: các mạch giải mã**

- giải mã là quá trình ngược lại với quá trình mã hóa. Nghĩa là từ một tổ hợp giá trị của nhóm mã n chữ số hệ 2 ta tìm lại được 1 trong N ký hiệu hoặc lệnh tương ứng.

**\* giải mã BCD sang thập phân**

- bảng chân lý cho trên bảng sau. Trong đó A,B,C,D là các đầu vào nhị phân  $L_0...L_9$  biểu diễn các thập phân từ  $0 \div 9$ . mỗi  $L_i = 1$  một lần ở đường chéo bảng, còn lại bằng 0 cả.

| Mã BCD |   |   |   | Mã thập phân |       |       |       |       |       |       |       |       |       |
|--------|---|---|---|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| D      | C | B | A | $L_0$        | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ | $L_7$ | $L_8$ | $L_9$ |
| 0      | 0 | 0 | 0 | 1            | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0      | 0 | 0 | 1 | 0            | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0      | 0 | 1 | 0 | 0            | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0      | 0 | 1 | 1 | 0            | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0      | 1 | 0 | 0 | 0            | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     |
| 0      | 1 | 0 | 1 | 0            | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     |
| 0      | 1 | 1 | 0 | 0            | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| 0      | 1 | 1 | 1 | 0            | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     |
| 1      | 0 | 0 | 0 | 0            | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     |
| 1      | 0 | 0 | 1 | 0            | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     |

⇒ Từ bảng chân lý ta suy ra hàm logic lối ra:

$$\begin{aligned}
 L_0 &= & L_5 &= AC \\
 L_1 &= A & L_6 &= BC \\
 L_2 &= B & L_7 &= ABC \\
 L_3 &= AB & L_8 &= D \\
 L_4 &= C & L_9 &= D
 \end{aligned}$$

- Từ hàm logic ta có thể xây dựng sơ đồ giải mã này khi dùng 4 mạch NOT và 10 mạch AND lối vào. Trong thực tế người ta chế tạo sẵn những bộ giải mã này được tích hợp trong IC 7441 có 16 chân hoặc vi mạch 74LS145

**\* giải mã BCD sang mã 7 đoạn**

- mã nhị phân BCD được chuyển sang thập phân và hiển thị các số thập phân bằng ma trận 7 đoạn. 7 đoạn sáng này có thể là led ( hoặc tinh thể long). Ứng với mỗi tổ hợp xác định các thanh sáng sẽ hiển thị cho ta một chữ số trong hệ 10.

- bảng chân lý quan hệ giữa lối vào nhị phân A,B,C,D và lối ra là các tổ hợp thanh sáng từ a đến g hiển thị các số của hệ 10 từ 0 ÷ 9 trong bản sau:

| BCD |   |   |   | 7 đoạn |   |   |   |   |   |   |
|-----|---|---|---|--------|---|---|---|---|---|---|
| D   | C | B | A | a      | b | c | d | e | f | g |
| 0   | 0 | 0 | 0 | 1      | 1 | 1 | 1 | 1 | 1 | 0 |
| 0   | 0 | 0 | 1 | 0      | 1 | 1 | 0 | 0 | 0 | 0 |
| 0   | 0 | 1 | 0 | 1      | 1 | 0 | 1 | 1 | 0 | 1 |
| 0   | 0 | 1 | 1 | 1      | 1 | 1 | 1 | 0 | 0 | 1 |
| 0   | 1 | 0 | 0 | 0      | 1 | 1 | 0 | 0 | 1 | 1 |
| 0   | 1 | 0 | 1 | 1      | 0 | 1 | 1 | 0 | 1 | 1 |
| 0   | 1 | 1 | 0 | 0      | 0 | 1 | 1 | 1 | 1 | 1 |
| 0   | 1 | 1 | 1 | 1      | 1 | 1 | 0 | 0 | 0 | 0 |
| 1   | 0 | 0 | 0 | 1      | 1 | 1 | 1 | 1 | 1 | 1 |
| 1   | 0 | 0 | 1 | 1      | 1 | 1 | 0 | 0 | 1 | 1 |

- nhìn vào bảng chân lý ta thấy rằng giá trị của 7 hàm a ... g có giá trị là mức 1 quá nhiều. để đơn giản hàm logic ta viết biểu thức dưới dạng các hàm ÷ có giá trị mức 1 tức a÷ g có giá trị mức 0.

**\* giải mã johnson sang thập phân.**

- giả sử dùng 1 tổ hợp 5 bit mã Johnson để giải mã ra 10 số của hệ 10 từ 0÷ 9 ta được bảng chân lý như sau:

| số thập phân | J <sub>5</sub> | J <sub>4</sub> | J <sub>3</sub> | J <sub>2</sub> | J <sub>1</sub> | L <sub>0</sub> | L <sub>1</sub> | L <sub>2</sub> | L <sub>3</sub> | L <sub>4</sub> | L <sub>5</sub> | L <sub>6</sub> | L <sub>7</sub> | L <sub>8</sub> | L <sub>9</sub> |
|--------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0            | 0              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 1            | $\bar{1}$      |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 2            | $\bar{1}$      |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 3            | 1              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 4            | 1              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 5            | 1              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 6            | 0              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 7            | 0              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 8            | 0              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| 9            | 0              |                |                |                |                |                |                |                |                |                |                |                |                |                |                |

- Để giải mã Johnson ra thập phân từ 0,1...9 ta phải dùng 5 bit cho mã Johnson. Còn các số thập phân tương ứng là  $L_0 \div L_9$  nếu chọn đường chéo  $L_0 = "0"$  thì đèn sáng, thể hiện số  $i$  trong hệ thập phân thì ta thiết kế sơ đồ giải mã dùng cửa NAND. (nếu lựa chọn đường chéo là mức "1" thì đèn sáng ta sẽ thiết kế mạch giải mã cửa NOT )

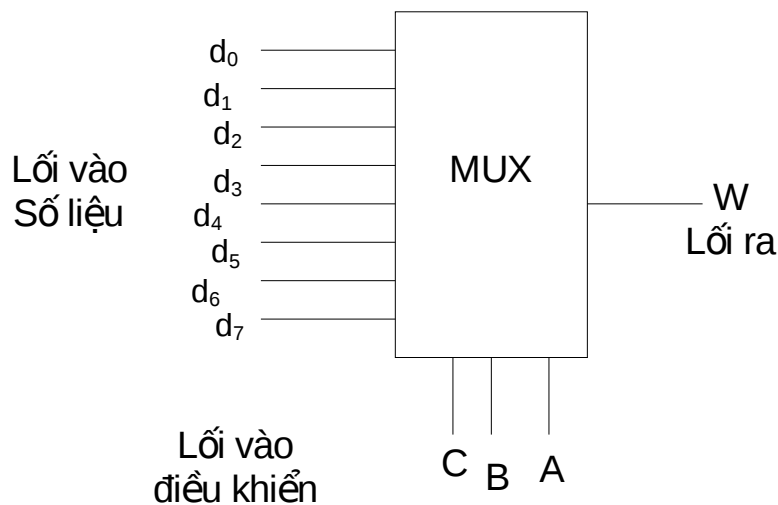
### Câu 7: mạch hợp kênh và phân kênh

#### \* mạch hợp kênh

- hợp kênh là một sơ đồ logic tổ hợp nhiều lối vào và 1 lối ra duy nhất. nó làm nhiệm vụ chọn lọc và truyền số liệu từ một trong những lối vào để đưa ra. Các lối vào ra của bộ hợp kênh gồm có:

- + các lối vào số liệu
- + các lối vào điều khiển

- tùy theo tổ hợp các giá trị lối vào điều khiển, sẽ là lệnh cho số liệu ở lối vào nào được truyền ra lối ra. nếu số lối vào điều khiển  $s$  tối thiểu là  $S = \log_2 n$ . ví dụ xây dựng bộ hợp kênh 8 lối vào 1 lối ra, như vậy sẽ cần 3 lối vào là A,B,C như hình sau:



- ta có thể xây dựng bảng chân lý cho hợp kênh này trên bảng sau:

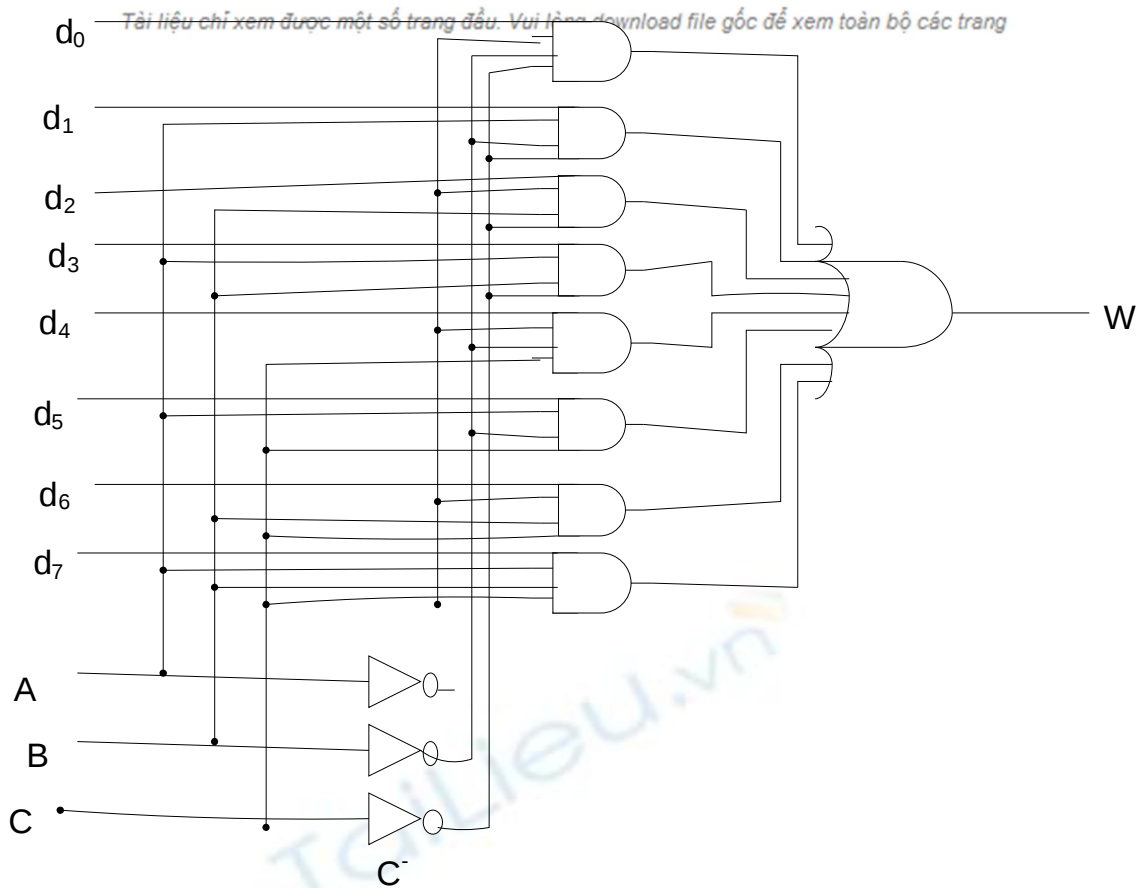
| điều khiển |   |   | lối vào        |                |                |                |                |                |                |                | ra             |
|------------|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| C          | B | A | d <sub>0</sub> | d <sub>1</sub> | d <sub>2</sub> | d <sub>3</sub> | d <sub>4</sub> | d <sub>5</sub> | d <sub>6</sub> | d <sub>7</sub> | W              |
| 0          | 0 | 0 | d <sub>0</sub> | x              | x              | x              | x              | x              | x              | x              | d <sub>0</sub> |
| 0          | 0 | 1 | x              | d <sub>1</sub> | x              | x              | x              | x              | x              | x              | d <sub>1</sub> |
| 0          | 1 | 0 | x              | x              | d <sub>2</sub> | x              | x              | x              | x              | x              | d <sub>2</sub> |
| 0          | 1 | 1 | x              | x              | x              | d <sub>3</sub> | x              | x              | x              | x              | d <sub>3</sub> |
| 1          | 0 | 0 | x              | x              | x              | x              | d <sub>4</sub> | x              | x              | x              | d <sub>4</sub> |
| 1          | 0 | 1 | x              | x              | x              | x              | x              | d <sub>5</sub> | x              | x              | d <sub>5</sub> |
| 1          | 1 | 0 | x              | x              | x              | x              | x              | x              | d <sub>6</sub> | x              | d <sub>6</sub> |
| 1          | 1 | 1 | x              | x              | x              | x              | x              | x              | x              | d <sub>7</sub> | d <sub>7</sub> |

- ở đây các kí hiệu d<sub>0</sub>...d<sub>7</sub> là các thông tin logic ở đầu vào. các ô trống trong bảng chân lí ứng với các d<sub>i</sub> bất kỳ.

- từ bảng chân lí ta viết được hàm logic sau:

$$W = d_0 + Ad_1 + Bd_2 + ABd_3 + Cd_4 + Cd_5 + BCd_6 + ABCd_7$$

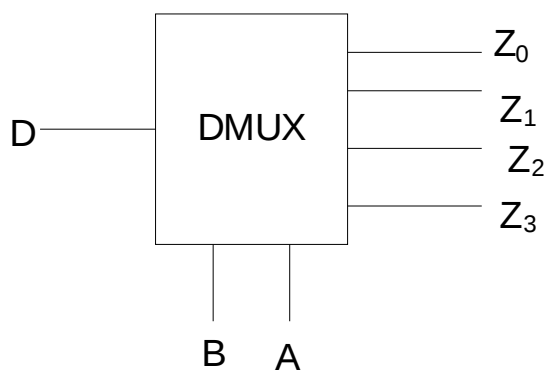
- từ phương trình ta nhận thấy, để thiết kế bộ hợp kênh này ta cần dùng 8 mạch AND 4 lối vào và 1 mạch OR 8 lối vào và 4 cửa ĐẢO như hình vẽ sau:



**\* mạch phân kênh.**

- bộ phân kênh có chức năng ngược với bộ hợp kênh. nó có 1 lối vào và nhiều lối ra, lối vào điều khiển cho phép thông tin đi từ lối vào chuyển ra một trong các lối ra nào đó.

ví dụ : ta xét bộ phân kênh 1 lối vào, 4 lối ra. như vậy cần 2 đầu điều khiển sơ đồ khối trên hình sau:



bảng chân lí như sau:

| B | A | Z <sub>0</sub> | Z <sub>1</sub> | Z <sub>2</sub> | Z <sub>3</sub> |
|---|---|----------------|----------------|----------------|----------------|
| 0 | 0 | D              | 0              | 0              | 0              |
| 0 | 1 | 0              | D              | 0              | 0              |
| 1 | 0 | 0              | 0              | D              | 0              |
| 1 | 1 | 0              | 0              | 0              | D              |



- Từ bảng chân lí ta viết được hàm logic.

$$Z_0 = D$$

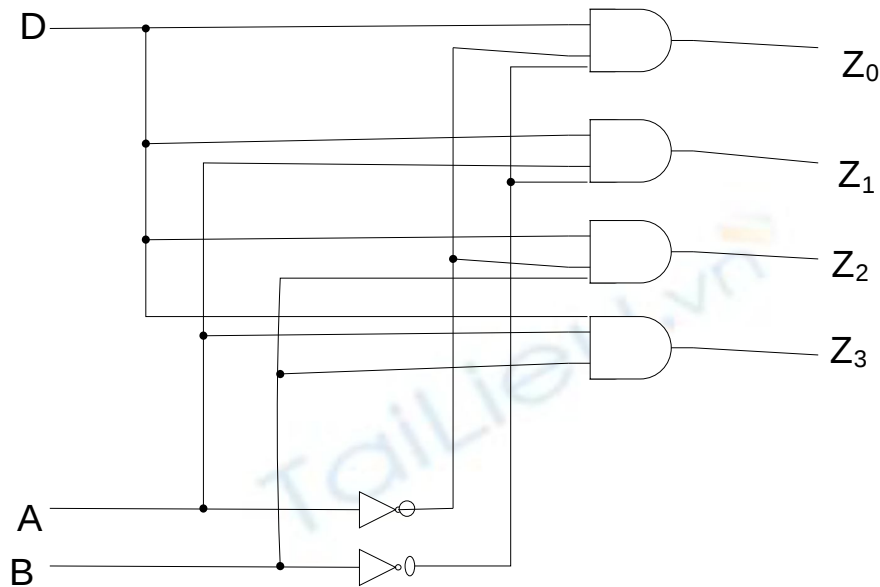
$$Z_1 = AD$$

$$Z_2 = BD$$

$$Z_3 = ABD$$

⇒ sơ đồ mạch gồm 2 cửa ĐẢO và 4 cửa VÀ 3 lối vào như hình sau. trong đó D là lối vào số liệu, A, B là 2 đầu vào có điều khiển cho phép số liệu D ra lối nào.

- Sơ đồ mạch phân kênh như sau:



## Câu 8: Các mạch số học

### I. Bộ cộng (Adder)

#### \* Bộ bán tổng (HA – Half Adder)

- Bộ bán tổng thực hiện cộng 2 số nhị phân 1 bit

Quy tắc cộng như sau:

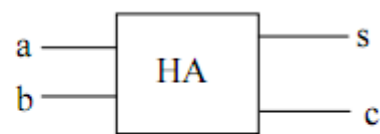
$$0 + 0 = 0 \text{ nhớ } 0$$

$$0 + 1 = 1 \text{ nhớ } 0$$

$$1 + 0 = 1 \text{ nhớ } 0$$

$$1 + 1 = 0 \text{ nhớ } 1$$

$$(a) \quad (b) \quad (s) \quad (c)$$



Hình 4.36. Mạch cộng 1 bit

Trong đó a, b là số cộng, s là tổng, c là số nhớ.

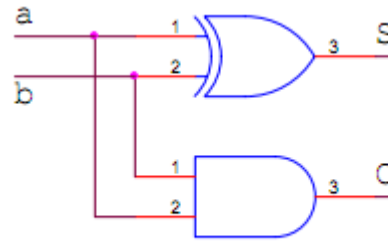
Bảng trạng thái mô tả hoạt động của mạch và phương trình logic :

$$s = a \cdot \bar{b} + \bar{a} \cdot b = a \oplus b$$

$$c = a \cdot b$$

Mạch cộng này chỉ cho phép cộng hai số nhị phân 1 bit mà không thực hiện cộng hai số nhị phân nhiều bit.

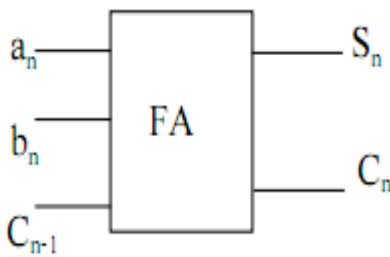
| a | b | s | c |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



Hình 4.37. Sơ đồ mạch cộng bán phần

**\* Bộ tổng (Bộ cộng toàn phần FA: Full Adder)**

Sơ đồ khối:



| a <sub>n</sub> | b <sub>n</sub> | C <sub>n-1</sub> | S <sub>n</sub> | C <sub>n</sub> |
|----------------|----------------|------------------|----------------|----------------|
| 0              | 0              | 0                | 0              | 0              |
| 0              | 1              | 0                | 1              | 0              |
| 1              | 0              | 0                | 1              | 0              |
| 1              | 1              | 0                | 0              | 1              |
| 0              | 0              | 1                | 1              | 0              |
| 0              | 1              | 1                | 0              | 1              |
| 1              | 0              | 1                | 0              | 1              |
| 1              | 1              | 1                | 1              | 1              |

Hình 4.38. Bộ cộng toàn phần

Trong đó:

- C<sub>n-1</sub>: Số nhớ của lần cộng trước đó
- C<sub>n</sub>: Số nhớ của lần cộng hiện tại
- S<sub>n</sub>: Tổng hiện tại

Từ bảng trạng thái mô tả hoạt động của mạch ta viết được phương trình logic:

$$S_n = f(a_n, b_n, C_{n-1})$$

$$C_n = f(a_n, b_n, C_{n-1})$$

Lập bảng karnaugh và tối thiểu hoá, ta có:

|                  |   | a <sub>n</sub> b <sub>n</sub> |    |    |    |
|------------------|---|-------------------------------|----|----|----|
|                  |   | 00                            | 01 | 11 | 10 |
| C <sub>n-1</sub> | 0 | 0                             | 1  | 0  | 1  |
|                  | 1 | 1                             | 0  | 1  | 0  |

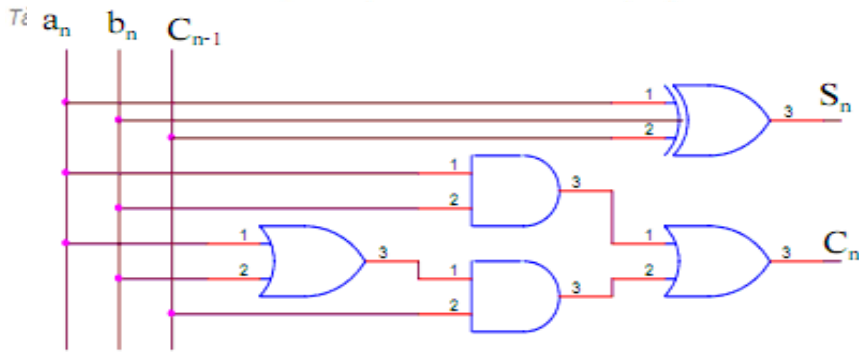
$$S_n = \overline{a_n} \overline{b_n} C_{n-1} + \overline{a_n} b_n \overline{C_{n-1}} + a_n \overline{b_n} \overline{C_{n-1}} + a_n b_n C_{n-1}$$

$$S_n = a_n \oplus b_n \oplus C_{n-1}$$

|                  |   | a <sub>n</sub> b <sub>n</sub> |    |    |    |
|------------------|---|-------------------------------|----|----|----|
|                  |   | 00                            | 01 | 11 | 10 |
| C <sub>n-1</sub> | 0 | 0                             | 0  | 1  | 0  |
|                  | 1 | 0                             | 1  | 1  | 1  |

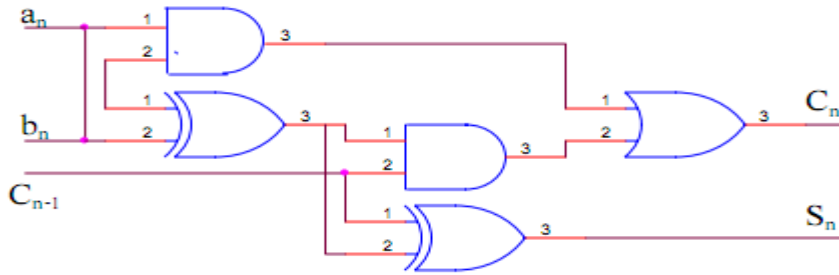
$$C_n = \overline{a_n} C_{n-1} + b_n C_{n-1} + a_n b_n$$

$$C_n = a_n b_n + C_{n-1} (a_n + b_n)$$



Hình 4.39. Mạch cộng toàn phần trực tiếp

Hoặc sử dụng HA để thực hiện FA:



Hình 4.40. Thực hiện mạch cộng toàn phần từ bộ bán tổng

## II. Bộ trừ (Subtractor)

### \* Bộ bán trừ (Bộ trừ bán phần – HS: Half subtractor)

Bộ bán trừ thực hiện trừ 2 số nhị phân 1 bit.

Quy tắc trừ như sau:

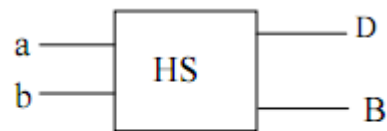
$$0 - 0 = 0 \text{ mượn } 0$$

$$0 - 1 = 1 \text{ mượn } 1$$

$$1 - 0 = 1 \text{ mượn } 0$$

$$1 - 1 = 0 \text{ mượn } 0$$

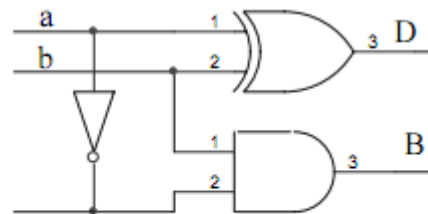
$$(a) \quad (b) \quad (D) \quad (B)$$



Hình 4.41 Mạch trừ bán phần

Trong đó a là số bị trừ, b là số trừ, D là hiệu, B là số mượn. Bảng trạng thái:

| a | b | D | B |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |



Hình 4.42. Sơ đồ logic

Phương trình logic:

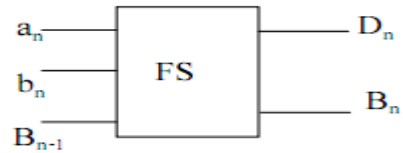
$$D = a \cdot \bar{b} + \bar{a} \cdot b = a \oplus b$$

$$B = \bar{a} \cdot b$$

Mạch này chỉ cho phép trừ hai số nhị phân 1 bit mà không thực hiện việc trừ hai số nhị phân nhiều bit.

**\* Bộ trừ toàn phần (FS – Full subtractor)**

| $a_n$ | $b_n$ | $B_{n-1}$ | $D_n$ | $B_n$ |
|-------|-------|-----------|-------|-------|
| 0     | 0     | 0         | 0     | 0     |
| 0     | 1     | 0         | 1     | 1     |
| 1     | 0     | 0         | 1     | 0     |
| 1     | 1     | 0         | 0     | 0     |
| 0     | 0     | 1         | 1     | 1     |
| 0     | 1     | 1         | 0     | 1     |
| 1     | 0     | 1         | 0     | 0     |
| 1     | 1     | 1         | 1     | 1     |



Hình 4.43. Mạch trừ toàn phần

Mạch có sơ đồ khối và bảng trạng thái mô tả hoạt động như trên:

Trong đó:  $B_{n-1}$ : Số mượn của lần trừ trước đó

$B_n$ : Số mượn của lần trừ hiện tại

$D_n$ : Hiệu số hiện tại

Lập bảng Karnaugh và tối thiểu hoá, ta có:

$D_n$  Karnaugh map:

|             |    |    |    |    |
|-------------|----|----|----|----|
| $a_n b_n$   | 00 | 01 | 11 | 10 |
| $B_{n-1}$ 0 | 0  | 1  | 0  | 1  |
| $B_{n-1}$ 1 | 1  | 0  | 1  | 0  |

$$D_n = \overline{a_n b_n} B_{n-1} + \overline{a_n} b_n \overline{B_{n-1}} + a_n \overline{b_n} \overline{B_{n-1}} + a_n b_n B_{n-1}$$

$$D_n = a_n \oplus b_n \oplus B_{n-1}$$

$B_n$  Karnaugh map:

|             |    |    |    |    |
|-------------|----|----|----|----|
| $a_n b_n$   | 00 | 01 | 11 | 10 |
| $B_{n-1}$ 0 | 0  | 1  | 0  | 0  |
| $B_{n-1}$ 1 | 1  | 1  | 1  | 0  |

$$B_n = \overline{a_n} B_{n-1} + b_n B_{n-1} + \overline{a_n} b_n$$

$$B_n = a_n b_n + B_{n-1} (a_n + b_n)$$

Có hai cách thực hiện bộ trừ toàn phần theo biểu thức logic đã tìm được: hoặc thực hiện trực tiếp (hình 4.44) hoặc sử dụng HS để thực hiện FS (hình 4.45).